

02-15-02

92173

Approved for use through 10/31/2002. OMB 0651-0031  
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/032,816
Filing Date	12/26/2001
First Named Inventor	Masateru Minemoto
Group Art Unit	2173
Examiner Name	Not assigned
Attorney Docket Number	056314/4

COPY OF PAPERS  
ORIGINALLY FILED

Total Number of Pages in This Submission **37**

## ENCLOSURES (check all that apply)

- ☐ Fee Transmittal Form
- ☐ Fee Attached
- ☐ Amendment / Reply
- ☐ After Final
- ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☒ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/Incomplete Application
- ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53

- ☐ Assignment Papers (for an Application)
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) \_\_\_\_\_

- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please identify below):

Remarks

RECEIVED

FEB 22 2002

Technology Center 2100

## SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Richard L. Moss, Esq. Reg. No. 39,782
Signature	
Date	02/13/2002

## CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: \_\_\_\_\_

Typed or printed name			
Signature		Date	

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



Enclosed Mail® mailing label

Number EL 781126572 US

Date of Deposit 02-13-02

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Richard L. Moss

Name

Signature



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年10月12日

出 願 番 号

Application Number:

特願2001-315109

出 願 人

Applicant(s):

嶺元 政輝

RECEIVED

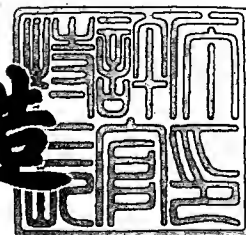
FEB 22 2002

Technology Center 2100

2001年12月21日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



【書類名】 特許願

【整理番号】 J-2001P001

【提出日】 平成13年10月12日

【あて先】 特許庁長官殿

【発明者】

【住所又は居所】 東京都大田区大森西4-10-11 田中マンション2  
03

【氏名】 嶺元 政輝

【特許出願人】

【識別番号】 500325908

【氏名又は名称】 嶺元 政輝

【代理人】

【識別番号】 100083851

【弁理士】

【氏名又は名称】 島田 義勝

【選任した代理人】

【識別番号】 100095533

【弁理士】

【氏名又は名称】 水谷 安男

【手数料の表示】

【予納台帳番号】 043786

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0011548

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 多次元プログラミング装置及び多次元プログラミング方法。

【特許請求の範囲】

【請求項 1】 2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクトファイル部と、

そのオブジェクトファイル部から読込まれたオブジェクト情報を用いて前記多次元流れ図を編集するオブジェクト編集部と、

編集された前記多次元流れ図を描画する描画部と、

編集された前記多次元流れ図を前記オブジェクトファイル部に保存する保存部からなることを特徴とする多次元プログラミング装置。

【請求項 2】 2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクトファイル部からオブジェクト情報を読み、

このオブジェクト情報を用いて前記多次元流れ図を編集し、

その編集された前記多次元流れ図に基づき、2次元流れ図、3次元流れ図、4次元流れ図等の多次元横書き流れ図を描画し、

また、編集された前記多次元流れ図を前記オブジェクトファイル部に保存することを特徴とする多次元プログラミング方法。

【請求項 3】 前記オブジェクトファイル部には、開始端子、終了端子等のプログラム流れ図記号情報、行と列によって特定される座標情報、前記行と列の交差区画としてのセル情報、文字情報が格納されていることを特徴とする請求項 1 に記載の多次元プログラミング装置。

【請求項 4】 前記座標情報には、時間軸、データ軸及び制御軸の組合わせ、時間軸、データ軸、制御軸及びCPU軸の組合わせ、時間軸、データ軸、制御軸、CPU軸及びイベント軸の組合わせ、時間軸、データ軸、制御軸、CPU軸、イベント軸及び条件軸の組合わせ並びに時間軸、データ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸の組合わせが含まれていることを特徴とする請求項 3 に記載の多次元プログラミング装置。

【請求項 5】 時間軸、データ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸を座標情報として描画される画面は、横軸を時間軸とし、縦軸をデータ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸とした画面構成であることを特徴とする請求項 4 に記載の多次元プログラミング装置。

【請求項 6】 前記オブジェクト編集部は、横軸に時間軸を、縦軸に少なくともデータ軸及び制御軸を取る三次元基本座標によって、プログラミング空間を画面に表示可能にすると共に、入力指令信号に従って、前記画面の編集等を行うことを特徴とする請求項 1 に記載の多次元プログラミング装置。

【請求項 7】 前記オブジェクト編集部は、前記プログラミング空間を切断してプログラムの内部を見るために、次元の切換えを可能とすることを特徴とする請求項 6 に記載の多次元プログラミング装置。

【請求項 8】 前記オブジェクト編集部は、縦軸をデータ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸とした画面構成によって、描画可能なプログラミング空間の平面をグループ化して、タブにより割当ることを特徴とする請求項 4 に記載の多次元プログラミング装置。

【請求項 9】 前記オブジェクト編集部は、行或いは列単位に座標情報を縮退或いは復元することを特徴とする請求項 3 に記載の多次元プログラミング装置。

【請求項 10】 前記オブジェクト編集部は、前記プログラミング空間に共通する時間軸を中心に、次元を低下させたごとくに、所定の座標軸を別の座標軸に埋め込むことを特徴とする請求項 3 に記載の多次元プログラミング装置。

【請求項 11】 前記保存部には、1 行分のセルのオブジェクトを保持する横スリット情報と、その横スリットに対応した平面上のオブジェクトを保持する平面オブジェクト情報が含まれることを特徴とする請求項 1 に記載の多次元プログラミング装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流

れ図を作成するための多次元プログラミング装置及び多次元プログラミング方法に関する。

【0002】

【従来の技術】

コンピュータを特定の目的に使用するためのプログラミングは、概ね、次の手順により作成される。

即ち、「ステップ1 プログラム流れ図→(変換)→ソースプログラム」, 「ステップ2 ソースプログラム→(コンパイラ)→オブジェクトプログラム」, 「ステップ3 オブジェクトプログラム→(リンカ)→ロードプログラム」になる。

【0003】

【発明が解決しようとする課題】

前記ステップ1におけるプログラム流れ図は、通常、図31に示したように、上から下へ、左から右へ、縦書きに記述される。

このような流れに従わない場合は、矢印を付けて流れの向きを明確にするようになっている。

即ち、平均値Hを求める図31のプログラム流れ図は、例えばディスプレイ上に、和S、入力値D、カウンタCの各データに対する制御の流れを時間軸に一致させて、縦方向に書継がれて表現される。

【0004】

しかし、この縦書きプログラム流れ図の作成には、次のような問題がある。

上述のように、データに対する制御の流れは縦の時間軸に沿うように書継がれているが、データ自体の流れは処理記号等の中で横方向に記述されている。しかも、データの流れは制御の流れによりその連続性が理解し難くなっている。

従って、縦書きプログラム流れ図の中に、縦方向の制御の流れと、横方向のデータの流れが混在し、データの流れが分断されている。

制御の流れが複雑になり、またデータの種別が多くなれば、この問題点がより顕著になる。

【0005】

一方、前記ステップ 1 におけるソースプログラムは一次元的に表現されるもので、制御の流れもデータの流れも寸断されて、前記縦書き流れ図よりも理解し難くなっている。

【0006】

そこで、本願発明は前記ステップ 1 の段階で行われている作業において、縦書きプログラム流れ図等の不都合を解消し、且つ、プログラミングの手法を一変させるような、多次元の横書き流れ図を作成するための多次元プログラミング装置及び多次元プログラミング方法を提供することを目的とする。

【0007】

【課題を解決するための手段】

この目的を達成するため、本願発明は、2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクトファイル部と、そのオブジェクトファイル部から読込まれたオブジェクト情報を用いて前記多次元流れ図を編集するオブジェクト編集部と、編集された前記多次元流れ図を描画する描画部と、編集された前記多次元流れ図を前記オブジェクトファイル部に保存する保存部からなることを特徴とする多次元プログラミング装置とした（請求項 1 に記載の発明）。

【0008】

また、2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクトファイル部からオブジェクト情報を読み、このオブジェクト情報を用いて前記多次元流れ図を編集し、その編集された前記多次元流れ図に基づき、2次元流れ図、3次元流れ図、4次元流れ図等の多次元横書き流れ図を描画し、また、編集された前記多次元流れ図を前記オブジェクトファイル部に保存することを特徴とする多次元プログラミング方法とした（請求項 2 に記載の発明）。

【0009】

ここで、発明者が本発明を想到するに至った、根底をなすアイデアを説明する

経験的に明らかなように、三次元の直方体は視点を変えることによって、色々



な形に見える。例えば正面から見た形と、平面から見た形と、側面から見た形が異なるごとくである。即ち、視点によって見えたり、見えなくなったりする部分があることを意味する。

このことを多次元直方体を想定した上で敷衍すれば、真上から見た図、真中から見た図、斜め上から見た図等のように、それぞれ各視点により見える部分と、見えない部分が出てくることを意味する。

各視点の位置は座標によって特定されるものであり、各視点により見える部分と、見えない部分が出てくるのは、座標軸が入れ替わって見えるからと考えることができる。

#### 【 0 0 1 0 】

次に、プログラムを多次元直方体になぞって考え、プログラミング空間なる概念を案出する。

まず、プログラムを構成する要素の時間、データ及び制御から 3 次元のプログラミング空間を考え、これをプログラミング基礎空間（3 次元基本座標ともいう）と定義する。

このプログラミング基礎空間を図示すれば、図 1 のよう示すことが出来る。

即ち、制御軸と流れ軸（時間軸）のなす平面を平面図 1、データ軸と流れ軸（時間軸）のなす平面を正面図 2、制御軸とデータ軸のなす平面を側面図 3 とする図法である。

#### 【 0 0 1 1 】

このようなプログラミング基礎空間に、例えばイベント処理が加わると四次元のプログラミングとなるが、これは図 2 のようにプログラミング基礎空間が重なったものとする。

この図 2 では、上の層からイベント処理 1、イベント処理 2、イベント処理 3、イベント処理 4、イベント処理 5 となっている。

この図 2 では、データ軸とイベント軸が重なって見えており、視点を変えると図 3 のように見える。

このようにプログラムを多次元直方体になぞって考えることができる。

#### 【 0 0 1 2 】

しかし、プログラムの編集の際に使用するディスプレイ等は二次元の表示装置であり、多次元空間そのものを可視化することは不可能である。

そこで、図 1 に示したプログラミング基礎空間を、展開して図 4 に示したように平面化することとする。

ここでは、制御流れ面 1 0 には、入出力と処理の流れが表示される。入出力の制御タイミングは制御の流れのポイントとなるので、この平面に記述する。

【 0 0 1 3 】

データ流れ面 2 0 には、データの流れが表示される。

データは入力装置から CPU を介してレジスタ等の内部メモリに入ることになるので、データの流れの始点になり、一方、データの出力は CPU を介して前記内部メモリから出力装置等へ送られるので、この送られる時点がデータの流れの終点になり、データの流れが表示される。

【 0 0 1 4 】

データ状態面 3 0 には、各処理の時点の内部メモリの状態を表示するようにする。

【 0 0 1 5 】

図 4 の横軸には、時間軸が取られおり、その時間軸にそって、制御フローと、データフローを表現する横書き流れ図を書くことができる。

【 0 0 1 6 】

前記データ流れ面 2 0 の垂直線で囲まれた部分 2 1 は、状態変化の集合を意味するもので、この状態の集合を前記データ状態面 1 2 に表示するものとする。これらの状態集合の要素が多い場合には、前記部分 2 1 をスクロールして表示されることになる。

よって、このデータ状態面 3 0 は、データを一時的に記憶するメモリの状態、例えば CPU に設けられた内部メモリの状態を表示できるようになっている。

未使用表示領域 4 は、ここをメモリ状態変化の集合を表示するデータ状態面に用いても良い。

なお、図 4 に示した太字 2 重線 7 0、7 1 は、後述するカーソルを想定している。

【 0 0 1 7 】

本願発明では、上記のようなプログラミング基礎空間をディスプレイの二次元表示装置に表現できるようにするため、2次元流れ図、3次元流れ図、4次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクトファイル部を備えている（請求項1及び2の各発明）。

前記オブジェクト情報には、開始端子、終了端子等のプログラム流れ図記号情報、行と列によって位置を特定するための座標情報、前記行と列の交差区画としてのセル情報、文字情報等が格納されている（請求項3の発明）。

【 0 0 1 8 】

多次元空間をディスプレイ等の二次元の表示装置で可視化するには、前記オブジェクト情報に座標情報を含ませることが重要であって、前記座標情報には、時間軸、データ軸及び制御軸の組合わせ、時間軸、データ軸、制御軸及びCPU軸の組合わせ、時間軸、データ軸、制御軸、CPU軸及びイベント軸の組合わせ、時間軸、データ軸、制御軸、CPU軸、イベント軸及び条件軸の組合わせ並びに時間軸、データ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸の組合わせが含まれていることが好ましい（請求項4の発明）。

プログラムは時間論理であることから、時間との組合わせがポイントとなり、このような座標情報により、プログラミング空間を平面化することができる。

なお、プログラムに関与するCPUは一台に仮定し、CPU軸は省略することにすれば、上記座標情報の内、時間軸、データ軸、制御軸及びCPU軸の組合わせもまた、前記プログラミング基礎空間となる。

【 0 0 1 9 】

次に問題となる点は、ディスプレイ等の画面に上記各軸をどのように割り振るかである。

本発明では、時間軸、データ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸を座標情報として描画される画面は、横軸を時間軸とし、縦軸をデータ軸、制御軸、CPU軸、イベント軸、条件軸及びPC軸とした画面構成であることを特徴とする（請求項5の発明）。

即ち、プログラミング空間に共通する時間軸を横軸に置くことにより、そのプ

プログラミング空間の平面化を的確に図ることができる。

【 0 0 2 0 】

上記発明の前記オブジェクト編集部は、横軸に時間軸を、縦軸に少なくともデータ軸及び制御軸を取る三次元基本座標によって、プログラミング空間を画面に表示可能にすると共に、入力指令信号に従って、前記画面の編集等を行うことを特徴とする（請求項 6 に記載の発明）。

即ち、ディスプレイ等の画面に横軸に時間軸を、縦軸に少なくともデータ軸及び制御軸が表示されるので、時間軸、データ軸及び制御軸によって特定される各セルに、入力指令信号に従って、オブジェクトの書込み等が行われる。

【 0 0 2 1 】

この場合、前記オブジェクト編集部は、前記プログラミング空間を切断してプログラムの内部を見るために、次元の切換えを可能とするようにする（請求項 7 に記載の発明）。

多次元直方体として想定される前記プログラミング空間の内部を見るには、所定の個所を切断すればよい。

例えば図 2 において、A-A で切断することにより、制御軸上のある時点の各イベント内容を把握できる。

このプログラミング空間の切断に対応するディスプレイ等の画面上の編集処理が次元の切換え、言い換えれば視点の位置の変更である。

この次元の切換えにより、多次元空間の内部の可視化を図ることができ、プログラミングの理解を促進させる。

【 0 0 2 2 】

また、前記オブジェクト編集部は、縦軸をデータ軸、制御軸、CPU 軸、イベント軸、条件軸及び PC 軸とした画面構成の場合、描画可能なプログラミング空間の平面をグループ化して、タブにより割当るようにする（請求項 8 に記載の発明）。

このようにすれば、ディスプレイ等の表示装置の画面が狭い場合、タブを導入することにより表示装置の平面を幾つかのグループに分けてタブに割り当てて、タブのクリックにより表示画面を切換え、画面を広く使うことができる。

【 0 0 2 3 】

同様に、表示装置の画面が狭い場合、プログラム全体の展望が困難になる。

そこで、前記オブジェクト編集部は、行或いは列単位に座標情報を縮退或いは復元する機能を備えている（請求項 9 に記載の発明）。

よって、プログラム全体の展望画面と細部の詳細画面を切換えることができる。

【 0 0 2 4 】

また、前記オブジェクト編集部は前記プログラミング空間に共通する時間軸を中心に、次元を低下させたごとくに、所定の座標軸を別の座標軸に埋め込む機能を備えている（請求項 1 0 に記載の発明）。

よって、限られた表示装置の画面の中で、次元を低下させることができる。

【 0 0 2 5 】

また、前記保存部には、1 行分のセルのオブジェクトを保持する横スリット情報と、その横スリットに対応した平面上のオブジェクトを保持する平面オブジェクト情報が含まれる（請求項 1 1 に記載の発明）。

プログラムは時間と密接に関係する論理であり、過去にどのような経路をたどり、将来どのような経路をたどる可能性があるかが容易に把握できるほうが良い。

その場合、横スリットは 1 行分のセルのオブジェクトを保持しており、これに対応するように平面オブジェクト情報は平面上のオブジェクトを保持しているので、画面上にプログラムの過去の経緯と将来の推移の写し込みが可能となる。

【 0 0 2 6 】

【発明の実施の形態】

以下、図面を参照しつつ、本願発明に係る実施形態を説明する。

実施形態に係る多次元プログラミング装置（以下、単に装置ともいう）は、図 5 に示したように、多次元の横書き流れ図作成用のオブジェクト情報が格納されたオブジェクトファイル部 5 及びプログラムファイル部 6 を備えたメモリ手段 7 と、前記オブジェクトファイル部 5 から読込まれたオブジェクト情報を用いて前記多次元流れ図を編集するオブジェクト編集部 8 及び編集された前記多次元流れ

図を前記オブジェクトファイル部 5 に保存する保存部 9 を備えた中央演算装置 (CPU) 9 A と、編集された前記多次元流れ図を描画する描画部としての画像処理部 9 B からなる。

【 0 0 2 7 】

前記装置は、キーボード、マウス、ディジタイザ等からなる入力装置 9 0 と、前記中央処理装置 9 A が加工したデータを一次的に保存する RAM 9 1 と、表示装置としての CRT 9 2 と、プリンタ等の出力装置 9 3 を備えている。

【 0 0 2 8 】

前記オブジェクトファイル部 5 は、オブジェクト ID、座標情報、セル情報、文字情報等が格納されている。

【 0 0 2 9 】

前記オブジェクト ID として、開始端子、終了端子、ループ開始端子、ループ終了端子、2 分岐判断、多分岐判断、合流、並列処理開始端子、並列処理終了端子、データ入力、データ出力、処理、定義済呼出等の、J I S 規格のプログラム流れ図記号情報を保持している。

マクロ化のためのマクロ命令開始端子、マクロ命令終了端子、マクロ分岐、マクロ合流などの ID を書込んでもよい。

【 0 0 3 0 】

前記座標情報として、時間軸、データ軸及び制御軸の組合わせ、時間軸、データ軸、制御軸及び CPU 軸の組合わせ、時間軸、データ軸、制御軸、CPU 軸及びイベント軸の組合わせ、時間軸、データ軸、制御軸、CPU 軸、イベント軸及び条件軸の組合わせ並びに時間軸、データ軸、制御軸、CPU 軸、イベント軸、条件軸及び PC 軸の組合わせが含まれている。

ここで、PC は、LAN に接続されているコンピュータのことである。

【 0 0 3 1 】

前記セル情報として、セルの横幅、即ち、前記座標情報の時間軸の間隔は、図記号の大きさに自動調整されるようになっている。なお、セルの縦幅は一定とする。その他、セルの色情報等が含まれている。

前記文字情報として、文字色、文字の背景色、文字列 (割当文、分岐条件等)

が格納されている。

その他、図記号の色情報、フロー検査フラッグ（0：未検査 1：検査終了）が格納されている。

#### 【0032】

なお、前記メモリ手段7に、短縮語辞書ファイル部70（図示せず）を設けてもよい。

表示装置92の表示画面の面積には限界があり、その横幅が狭いとき、関数名、変数名等の識別子が長いと、データフロー、制御フローの表示が短くなり、フローの理解が困難になる。そこで、現在作成中のプログラミングに固有の短縮語辞書を前記メモリ手段7に持つようにすることが好ましい。

そして、長い識別子を短くした短縮語を定義し、短縮語辞書に登録する。長い識別子の代りにこの短縮語を使うことにより、フローを長く表示することができ、プログラミングの理解が容易になる。

プログラミング固有の短縮語辞書のほか、システム全体で使用する短縮語辞書も用意してもよい。

#### 【0033】

さらに、前記メモリ手段7に、クラスライブラリ71（図示せず）を設けてもよい。

プログラミング言語C++等では、クラスという概念で、データとプログラムを緊密に関連付けている（カプセル化）。このようなオブジェクト指向プログラミングを本発明に係る多次元プログラミングに取り入れることが望ましい。

クラスの定義を図式化すると図6のようになる。

外部変数1、2、3、4…は、メンバ関数1、2、3、4…の外で定義されているので、外部変数になる。

従って、メンバ関数から自由にアクセスできる。この外部変数は、データ平面（後述、図4のデータ流れ面20に相当）に記述される。

メンバ関数の引数の個数は、各メンバ関数により異なる。また、引数、内部変数ともデータ平面に記述される。

派生クラスは、基本クラスから導出されるクラスである。基本クラスは導出元

のクラスである。メンバ関数をクリックすると多次元プログラミングの画面に移行し、メンバ関数の定義ができるようになる。

なお、メンバ関数の定義画面を図 7 示す。

この図は、制御平面（後述、図 4 の制御流れ面 1 0 に相当）とデータ平面を模倣的に示したものである。このメンバ関数 3 の定義が終わると、先のクラス定義に戻り、次のメンバ関数の定義に進むことになる。

#### 【 0 0 3 4 】

なお、前記プログラムファイル部 6 には、アプリケーションソフトとして、表計算ソフトのような多次元プログラミング用のソフトが格納されている。

#### 【 0 0 3 5 】

上記のように構成された装置による処理の流れを図 8 に示す。

即ち、前記中央処理装置 9 A が、2 次元流れ図、3 次元流れ図、4 次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクト MD ファイル（オブジェクトファイル部 5）からオブジェクト MD（オブジェクト情報）を読み込む。

次に、前記中央処理装置 9 A が、プログラムの入力指令に基づき、前記オブジェクト情報を用いて前記多次元流れ図を編集する。

そして、その編集された前記多次元流れ図に基づき、2 次元流れ図、3 次元流れ図、4 次元流れ図等の多次元横書き流れ図を描画し、前記表示装置 9 2 に表示する。

一方、前記中央処理装置 9 A は、編集された前記多次元流れ図を前記オブジェクトファイル部 5 に保存する。

#### 【 0 0 3 6 】

次に、上記装置による処理の流れをより具体的に説明する。

この装置が起動すると、前記表示装置 9 2 の画面にファイルメニューが表示される。

新規作成であれば、そのボタンを押し、次に次元の定義を行う。

次元は、横軸に時間軸を、縦軸に少なくともデータ軸及び制御軸を取る三次元基本座標を基本とし、この座標が前記表示装置 9 2 の画面に表示される。なお、



次元の増加は任意に行うことができ、例えば7次元の空間の場合には図9のような画面表示モデルとなる。

次にクラスの定義、関数の定義を行う。

【0037】

なお、表示装置92の表示画面の大きさが一定であるので、表示する次元が増えるほど、表示画面を見づらくなるが、このような場合、タブを導入して、図10に示したように平面を幾つかのグループに分けてタブに割り当てれば良い。

そして、前記表示装置92の表示画面の前記タブをクリックすると、その指令を受けた前記中央処理装置9Aが、表示画面を切換えるように制御する。

【0038】

次に、前記オブジェクトファイル部5から読み出したプログラム流れ図記号情報等に基づき、プログラマが各平面のセル内に流れ図記号等を描画する。

さらに、前記オブジェクトファイル部5から読み出した文字列情報から、文字列とオブジェクトとの関連を勘案して、セル内の所定の位置に文字列を描画する。

【0039】

画面上において、書込みを行う場所を示すカーソルは、スリット型十字カーソルとなっている。

例えば、図11のように4次元座標画面において、カーソルの存在する位置をカーソル座標とすると、ある時間cでのカーソル座標は、(時間c、データc、制御c、CPUc)の4つ数値の組で表されることになる。即ち、時間-CPU平面、時間-制御平面、時間-データ平面にカーソルが位置することになる。

【0040】

前記装置に従って、作成したプログラムを図12及び図13に示す。

なお、これらの図において、横軸の「S」は時間を、縦軸の「R」は制御を、「D」はデータを意味する。

【0041】

図12は、平均値を求めるプログラムの時間-制御平面の表示例を示すものである。この場合、同図の十字カーソル70のR2の位置で切断して見える場合の

時間-データ平面が図13のような表示となる。

【0042】

この図13において、S3は制御の流れに関与し、S5は判断であるので、データ流れ面20には表示されないことになる。S5の二重線は外部からの入力を示すもので、データフローの理解度を向上させるものである。

また、例えば、図12においてループ終点(S6)での側面図を表示させれば、図14のように、変数SとCの状態が表示される。

この平均値を求めるプログラムの実行時には具体的な数値が内部メモリに格納されることになるので、その側面図には数値が表示される。

【0043】

以上のプログラミング例によれば、プログラム流れ図の「制御の流れ」と「データの流れ」がきれいに整理され、理解し易い流れ図を表示できる。

【0044】

次に、以上の三次元基本座標に加えてイベント軸を加えた4次元プログラム流れ図について説明する。

前記図3において、プログラムの状態、即ち、制御の流れ線上のある時点におけるイベントとを組合わせると、図15のような表を描くことができる。

ここで、状態1の制御フローで切断すると、図16のようなデータフローが見える。即ち、状態1までのデータフロー、イベント処理1のデータフロー、イベント処理後のデータフローが連結性良く見える。

しかし、他のイベント処理のデータフローがこのままでは不連続となるので、イベント処理のデータフローを上下にずらすことにより、図17のように連結性が回復する。

しかし、制御フローを考えると、最上層のイベント処理1の制御フローは見えるが、その下層のイベント処理は全て隠れて見えない。

【0045】

このような不都合を解消するため、図18のような画面構成とする。同図において、状態1のイベント1の部分をクリックすると、そのクリック信号に基づき、イベント2の処理の制御フロー図を表示できるようにする。即ち、クリックす

るたびに、次のイベントの制御フロー図のデータを呼出して、画面に表示できるようにする。

これらの動作は、前記図 2 に示したプログラミング空間をイベント軸に直角に、切断したことを意味する。

即ち、状態 1 で切断し、イベント 2 と 3 の間で切断したデータフロー図は図 1 9 のように示されるが、これを真上から見る場合として図 2 0 のように表示させる。

#### 【 0 0 4 6 】

イベント処理の制御フローが大きくなると、状態 1、状態 2、状態 3 の間隔が広がり、処理の全体が見ずらくなる。そこで、図 2 1 (イ)、(ロ) のようにプログラミング基礎空間のイベント処理部の制御フローを縮退させ、必要に応じて拡張させることが望ましい。

これにより、状態 1、状態 2、状態 3 の間隔を狭めることができるので、プログラムの制御フローの全体が良くわかるようになる。

即ち、画面表示としては、図 2 2 が縮退表示、図 2 3 が拡張表示となる。

#### 【 0 0 4 7 】

上記イベント処理の拡張と縮退は、図 2 4、図 2 5 のような画面表示でも行うことができる。

前記表示装置 6 の画面が狭い場合、処理全体の展望が困難になるが、上記縮退と復元（拡張）の機能があると、全体の展望と細部の詳細を切換えて行うことができる。

図 2 4 は、イベント処理の元の画面であり、制御 1 から制御 1 2 までの行が連続して表示されている。

図 2 5 は、イベント処理を縮退したもので、前記図 2 4 の画面のうち、制御 4、5、8、1 0、1 1 の各行が縮退され、その縮退された分、新しい行が追加表示され、全体が展望し易くなっている。

#### 【 0 0 4 8 】

前記プログラム基礎空間を表計算ソフトのスプレッドシート形式にそって表すと、3 次元の場合は、図 2 6 のように、次元を低下させると図 2 7 のように表示

できる。

プログラムの時間一制御平面部を剥ぎ取ったものが、図 2 6 で、これは、二次元の表が重なったようになって 3 次元化されている。

この表を図 2 7 のように時間を共通にして順番に平面に並べると、二次元化となる。

このことは、時間一制御平面に、時間一データ平面を埋め込むことができることを意味する。

#### 【 0 0 4 9 】

次に、4 次元の画面を二次元に次元を低下させ、埋め込む場合を説明する。

図 2 7 の表が CPU 軸にそって並べられたものを想定する。

この画面は、もともと 4 次元空間（時間、データ、制御、CPU）のセルの集まりであり、この空間から三次元セル空間（時間、データ、制御）を選択し、これを二次元化（時間、データと制御）したものが、整列した 3 次元の表となり、この表を二次元化すると、図 2 8 の表に表示される。

4 次元以上の多次元画面も、同様な手順により、2 次元化することができる。

なお、以上の操作と逆の操作を行うことにより、もとの次元に戻すことができる。

#### 【 0 0 5 0 】

プログラムは、時間と密接に関係する論理であり、過去にどのような経路をたどり、将来どのような経路をたどる可能性があるかが容易に把握できることが好ましい。

そのため、プログラムをゴムのような柔らかいものからできているものと見做して、時間軸に垂直な方向へは自由に変形できるものとする。

過去の経路を保持する場所としてスリット状のカーソルが便利である。このことを図 2 9 の平面一 CPU 画面で説明する。

カーソル位置は（時間 4、CPU 1）である。ここで CPU 1 にあるスリットを CPU スリットと称する。

CPU スリットは、1 行分のセルのオブジェクトを保持するオブジェクトになる。

一方、時間についての一系列分のセルの情報を保持するオブジェクトを時間スリットとする。

【 0 0 5 1 】

同図において、CPUスリットには、座標情報、稼働可能情報、平行処理開始情報、同処理終了情報を示すオブジェクトIDが含まれている。

前記座標情報として、(時間1、CPU1)、(時間2、CPU1)、(時間3、CPU1)、(時間4、CPU1)、(時間5、CPU1)、(時間6、CPU1)、(時間7、CPU1)を持っている。

また、稼働可能情報として(時間1、CPU1)から(時間7、CPU1)での稼働可能オブジェクトIDが含まれている。

また、平行処理開始情報として、(時間3、CPU1)のオブジェクトIDを持ち、平行処理終了情報として(時間6、CPU1)のオブジェクトIDを持つ。

【 0 0 5 2 】

次に、図30のように前記カーソルの位置を(時間4、CPU2)に下げる。

同図において、点線はCPUスリットへの写し込みのために、見かけ上、空になっていることを示す。

座標(時間1、CPU2)、(時間2、CPU2)、(時間6、CPU2)と(時間7、CPU2)のセルは、実際、空である。

前記空のスリットにはCPUスリットのオブジェクトをそのまま表示する。

座標(時間3、CPU2)のセルは、並列処理開始のオブジェクトIDであるので、(時間3、CPU1)、(時間3、CPU2)のセルに対応したオブジェクトを表示し、(時間3、CPU2)オブジェクトをCPUスリットに保存する。

座標(時間4、CPU2)、(時間5、CPU2)のセルは、空ではないので、CPUスリットの時間4、5に保存表示する。

座標(時間4、CPU1)、(時間5、CPU1)のセルは、稼働可能のオブジェクトIDになっているので、対応するオブジェクトを描画しなおす。

座標(時間6、CPU2)のセルは、並列処理終了のオブジェクトIDである

ので、対応するCPUスリットに保存のオブジェクトを表示する。

座標（時間6、CPU1）のセルは、見かけ上、空に描画し直す。

座標（時間7、CPU2）のセルは、空であり、空のときはCPUスリットをそのまま表示する。

座標（時間7、CPU1）のセルは、画面上で見かけ上、空とする。以上の処理により、CPUスリットには、過去の経過と将来の推移が写し込まれる。

#### 【0053】

上記のような操作により編集されたプログラムは、ファイル名が付けられた後、前記オブジェクトファイル部5に保存される。

#### 【0054】

##### 【発明の効果】

本願発明によれば、縦書きプログラム流れ図等の不都合を解消し、且つ、プログラミングの手法を一変させるような、多次元の横書き流れ図を作成するための多次元プログラミング装置及び多次元プログラミング方法を提供することができる。

##### 【図面の簡単な説明】

【図1】 本願発明の前提となるプログラミング基礎空間の斜視図、

【図2】 同プログラミング基礎空間の斜視図、

【図3】 同正面図、

【図4】 同プログラミング基礎空間の展開図、

【図5】 本願発明に係る多次元プログラミング装置の構成例図、

【図6】 クラスの定義を図式化した説明図、

【図7】 メンバ関数の定義画面図、

【図8】 本願発明に係る制御フロー図、

【図9】 本願発明に係る多次元プログラミング装置の画面表示例図、

【図10】 同画面表示例図、

【図11】 同画面表示カーソル説明図、

【図12】 3次元プログラム流れ図の表示例図、

【図13】 3次元プログラム流れ図の表示例図、

- 【図 1 4】 3 次元プログラム流れ図の表示例図、
- 【図 1 5】 四次元のプログラミング基礎空間の切断図、
- 【図 1 6】 四次元のプログラミング基礎空間の切断図、
- 【図 1 7】 四次元のプログラミング基礎空間の切断図、
- 【図 1 8】 四次元の画面表示例図、
- 【図 1 9】 四次元のプログラミング基礎空間の切断図、
- 【図 2 0】 四次元の画面表示例図、
- 【図 2 1】 (イ) (ロ) 四次元のプログラミング基礎空間の縮退図と拡張

図、

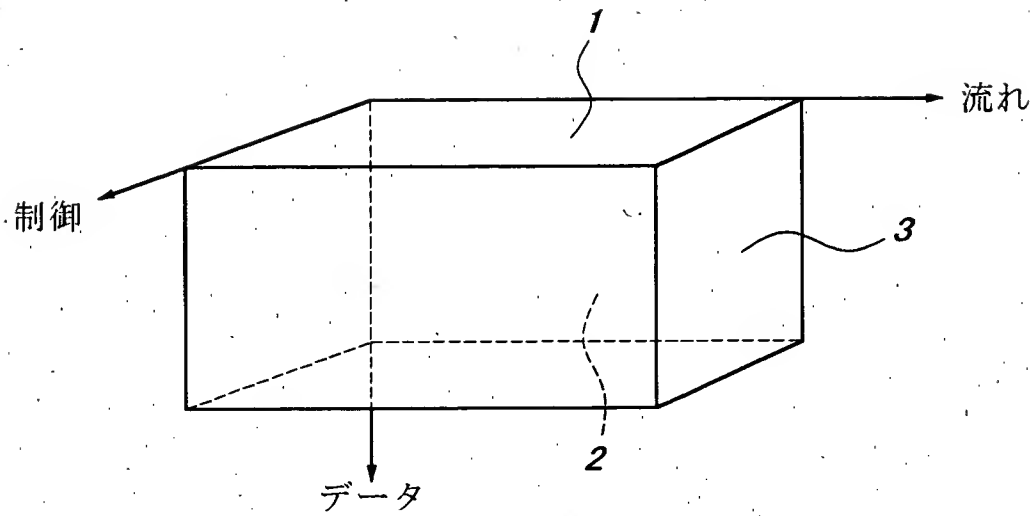
- 【図 2 2】 四次元の画面表示例図、
- 【図 2 3】 四次元の画面表示例図、
- 【図 2 4】 四次元の拡張の画面表示例図、
- 【図 2 5】 四次元の縮退の画面表示例図、
- 【図 2 6】 プログラミング基礎空間の埋込みの説明図、
- 【図 2 7】 プログラミング基礎空間の埋込みの説明図、
- 【図 2 8】 四次元の埋込みの画面表示例図、
- 【図 2 9】 データの写し込みの画面表示例図、
- 【図 3 0】 データの写し込みの画面表示例図、
- 【図 3 1】 従来のプログラミングの画面表示例図。

【符号の説明】

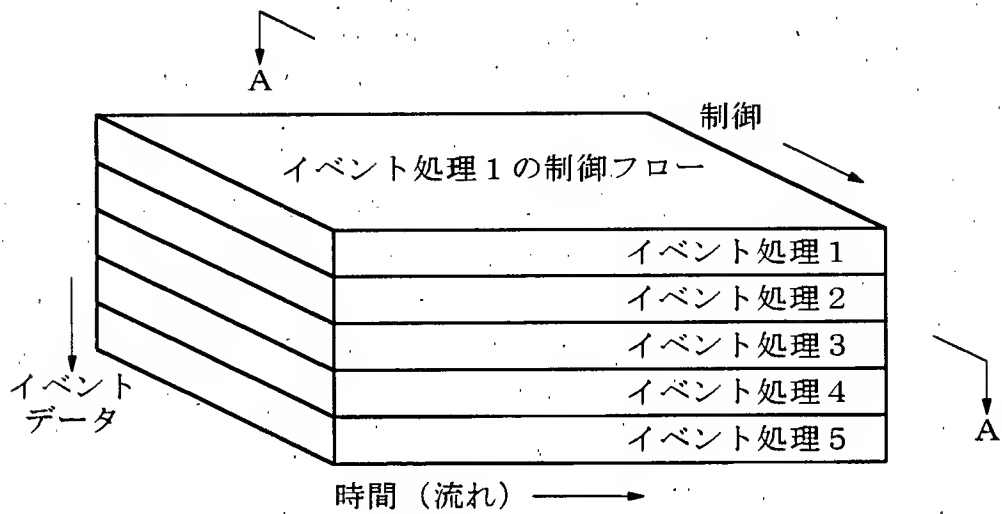
- |               |                    |
|---------------|--------------------|
| 5 オブジェクトファイル部 | 6 プログラムファイル部       |
| 7 メモリ手段       | 8 オブジェクト編集部        |
| 9 保存部         | 9 A 中央演算装置 (C P U) |
| 9 B 画像処理部     |                    |
| 9 0 入力装置      | 9 1 R A M          |
| 9 2 C R T     | 9 3 出力装置           |

【書類名】 図面

【図1】

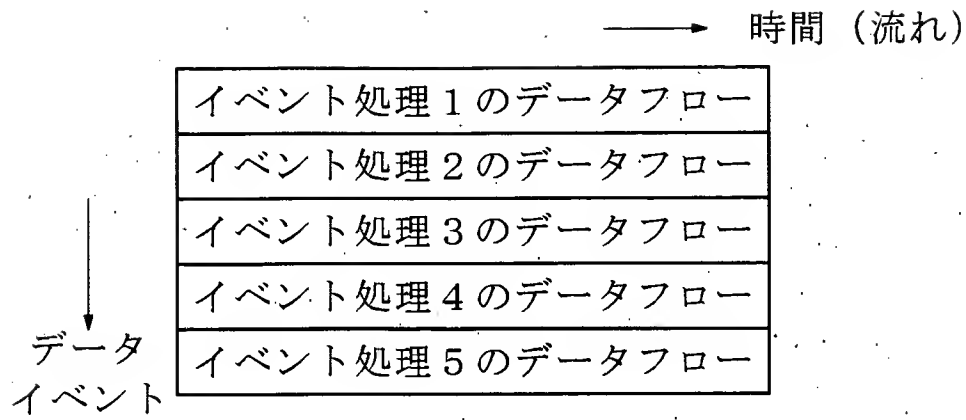


【図2】

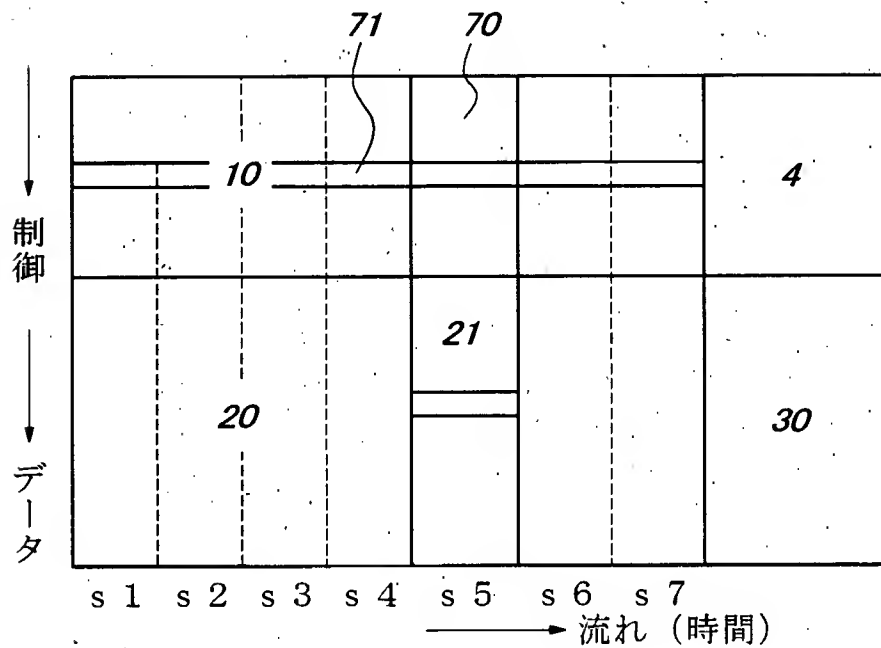




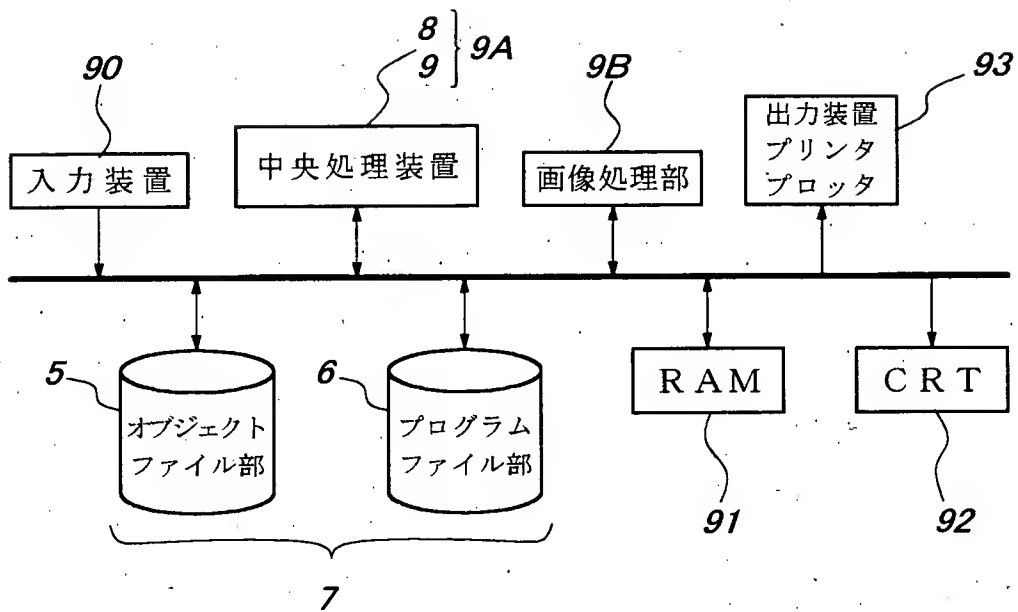
【図 3】



【図 4】



【図5】



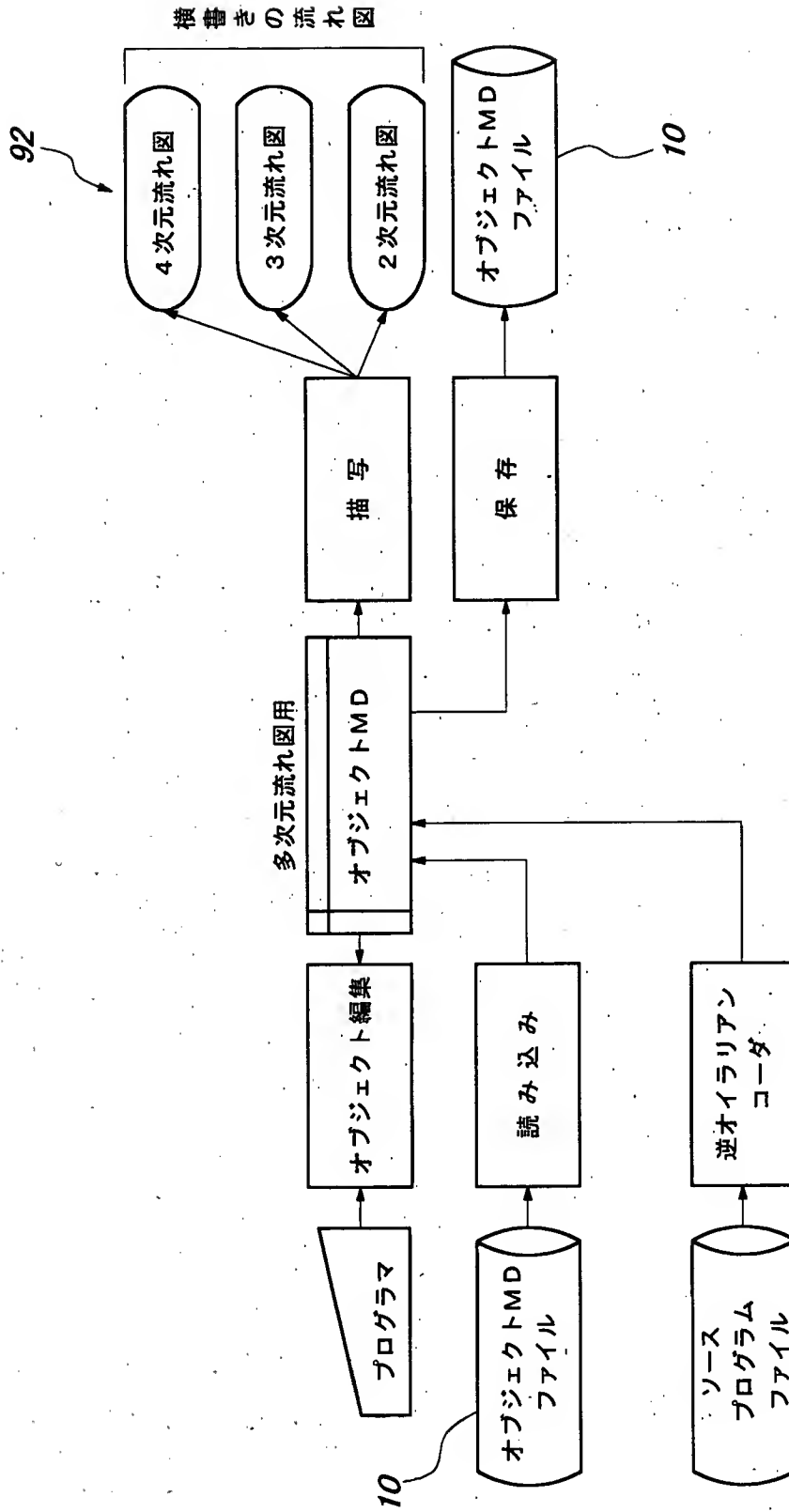
【図6】

派生クラス名			基本クラス名				
アクセス型1	型1	外部変数1	アクセス型A	型A	メンバ関数1	(型a	引数a, . . . )
アクセス型2	型2	外部変数2	アクセス型B	型B	メンバ関数1	(型b	引数b, . . . )
アクセス型3	型3	外部変数3	アクセス型C	型C	メンバ関数1	(型c	引数c, . . . )
アクセス型4	型4	外部変数4	アクセス型D	型D	メンバ関数1	(型d	引数d, . . . )
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

【図7】

型C メンバ関数3					
型1 外部変数名1					
型2 外部変数名2					
型3 外部変数名3					
型4 外部変数名4					
・ ・ ・					
型c 引数c					
・ ・ ・					
型t 内部変数t					
・ ・ ・					

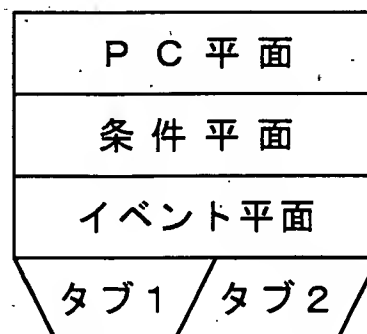
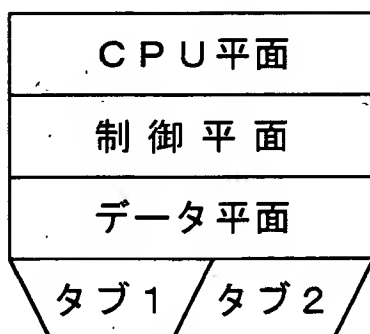
【図8】



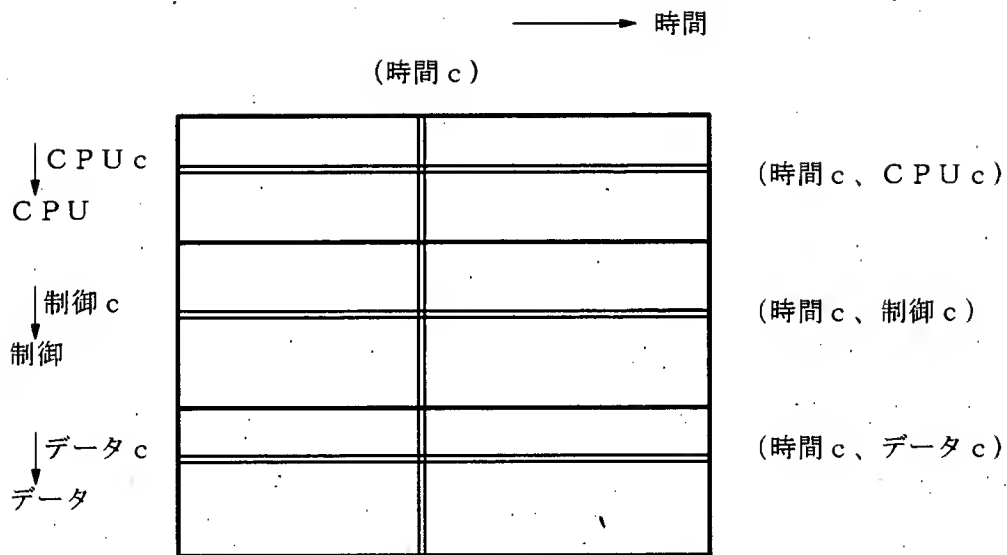
【図 9】

	時間1	時間2	時間3	時間4	時間5	時間6	時間7			
PC1										
PC2										
PC3										
PC4										
条件1										
条件2										
条件3										
条件4										
イベント1										
イベント2										
イベント3										
イベント4										
イベント5										
CPU1										
CPU2										
CPU3										
制御1										
制御2										
制御3								制御3	制御2	制御1
データ1										
データ2										
データ3										
データ4										
データ5										

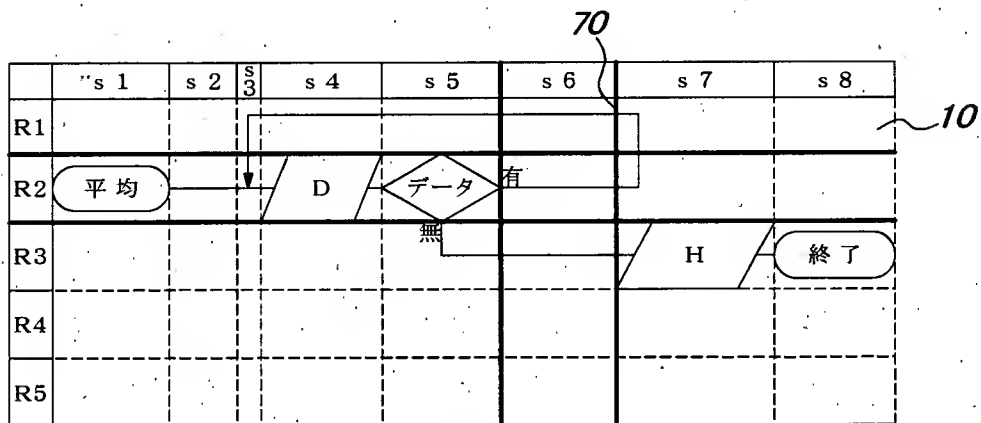
【図 1 0】



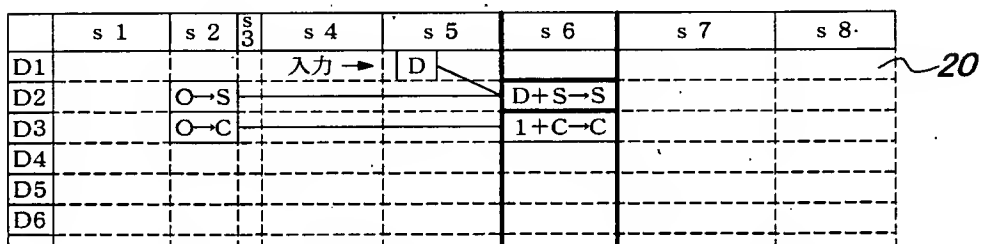
【図 1 1】



【図 1 2】



【図 1 3】



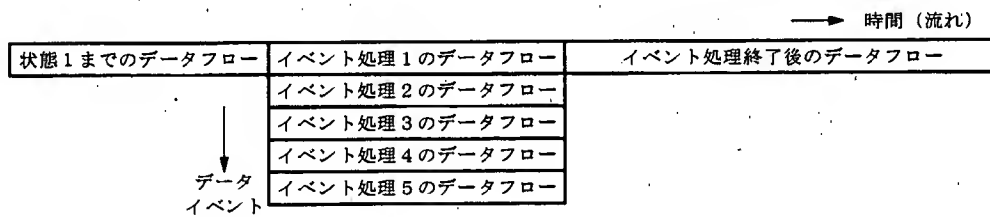
【図 14】

R 4	R 3	R 2	R 1	
		D		D 1
		S		D 2
	H	C		D 3
				D 4
				D 5
				D 6

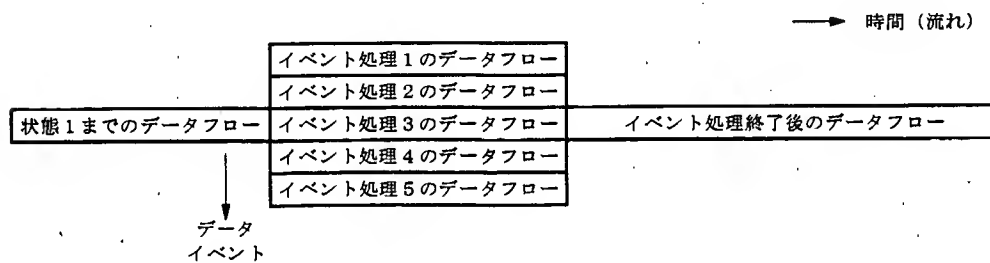
【図 15】

	イベント 1	イベント 2	イベント 3	イベント 4	イベント 5
状態 1					
状態 2					
状態 3					

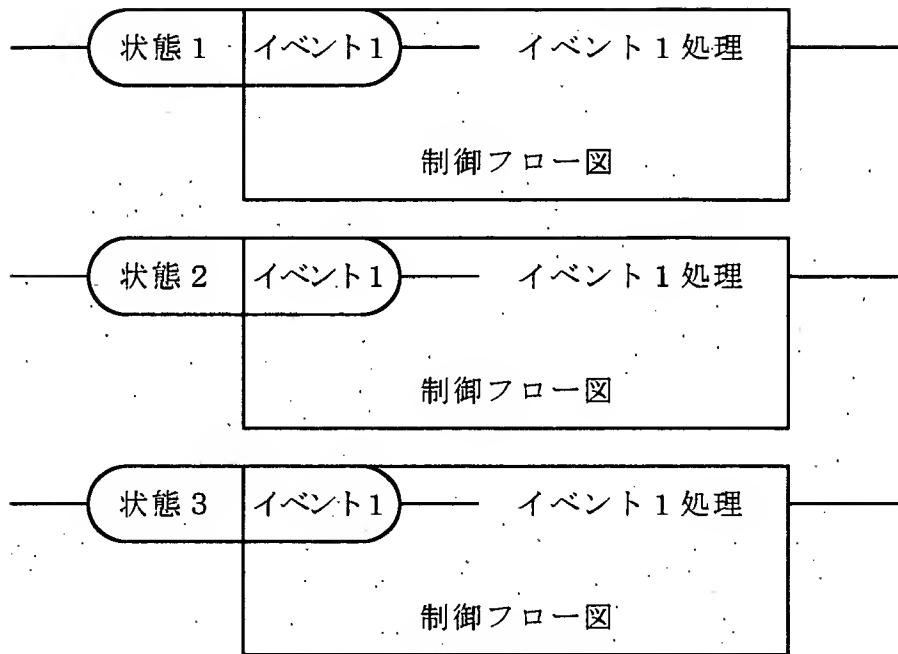
【図 16】



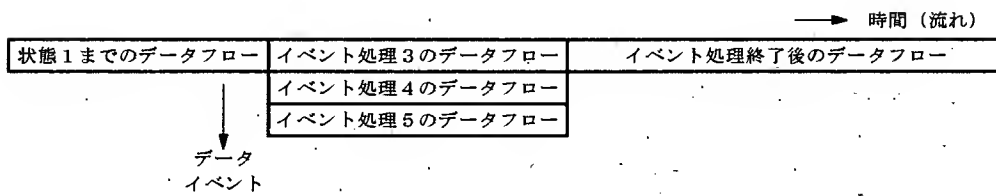
【図 17】



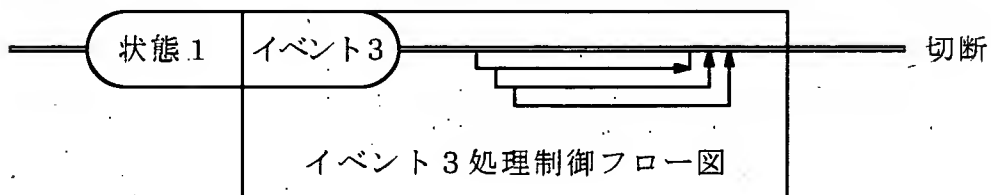
【図 1 8】



【図 1 9】

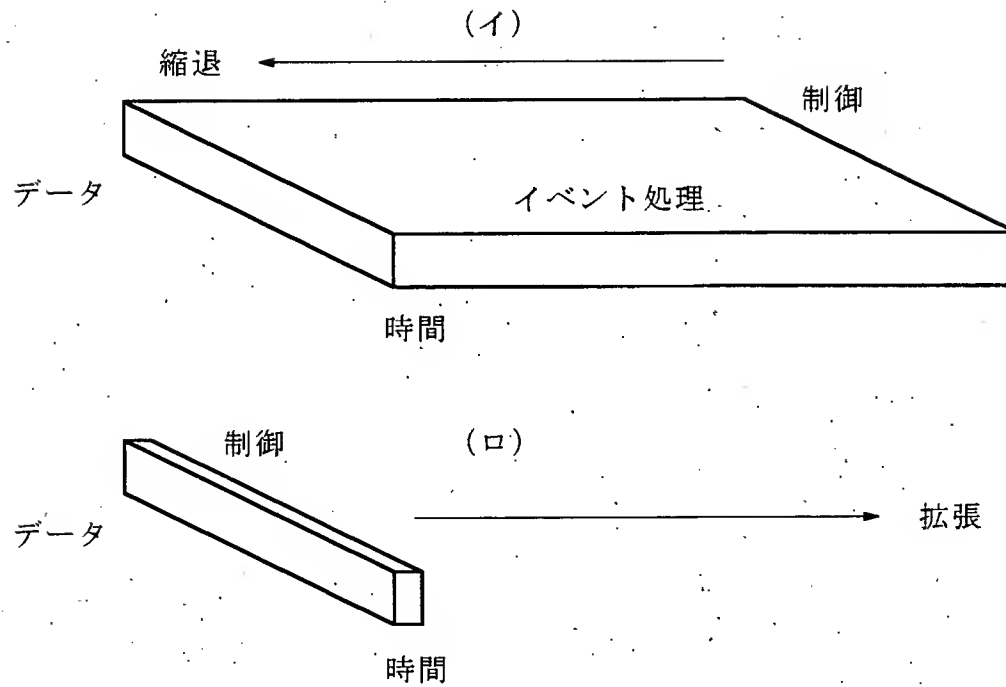


【図 2 0】

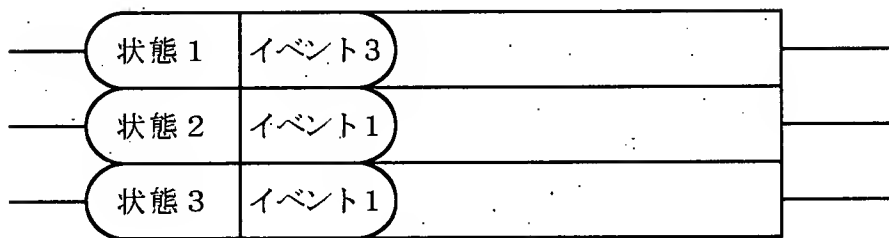




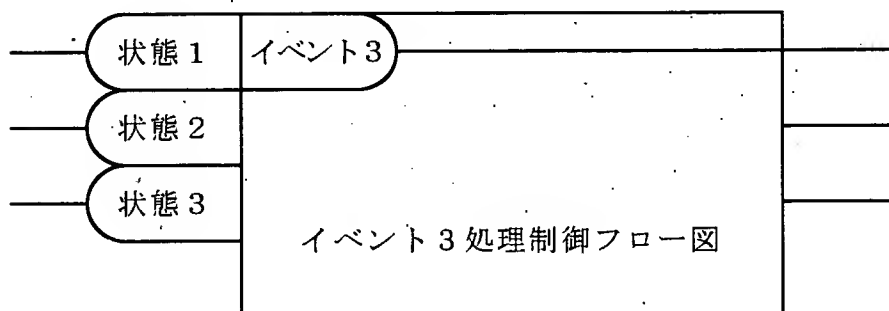
【図 2 1】



【図 2 2】



【図 2 3】



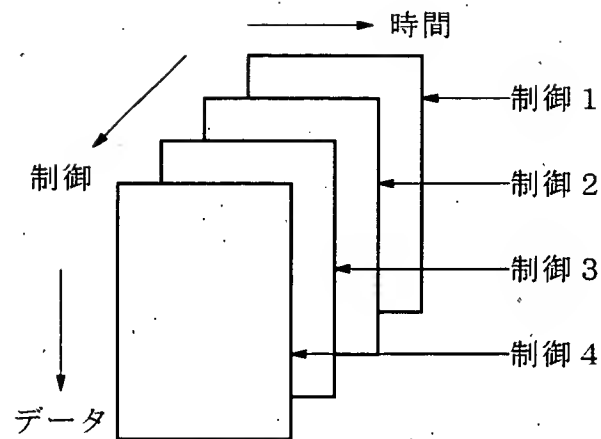
【図 24】

[illegible]

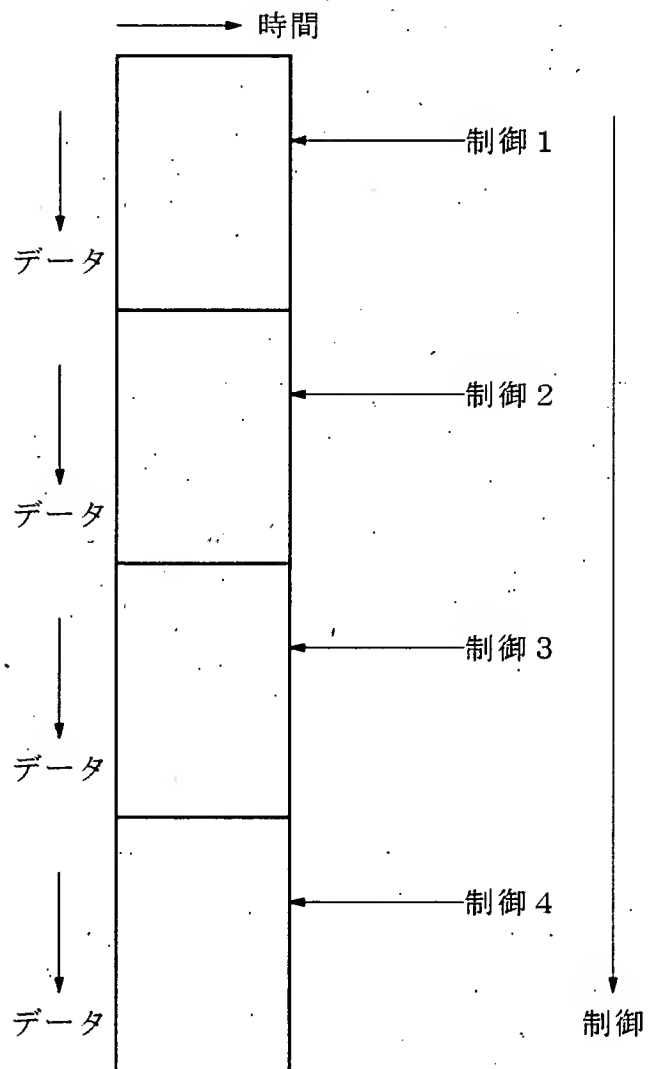
【图 2-5】

[illegible]

【図 26】



【図 27】



【図 28】

		時間1	時間2	時間3	時間4	時間5	時間6	時間7	時間8
CPU1	制御1								
		データ1							
		データ2							
	制御2								
		データ1							
		データ2							
	制御3								
		データ1							
		データ2							
CPU2	制御1								
		データ1							
		データ2							
	制御2								
		データ1							
		データ2							
	制御3								
		データ1							
		データ2							

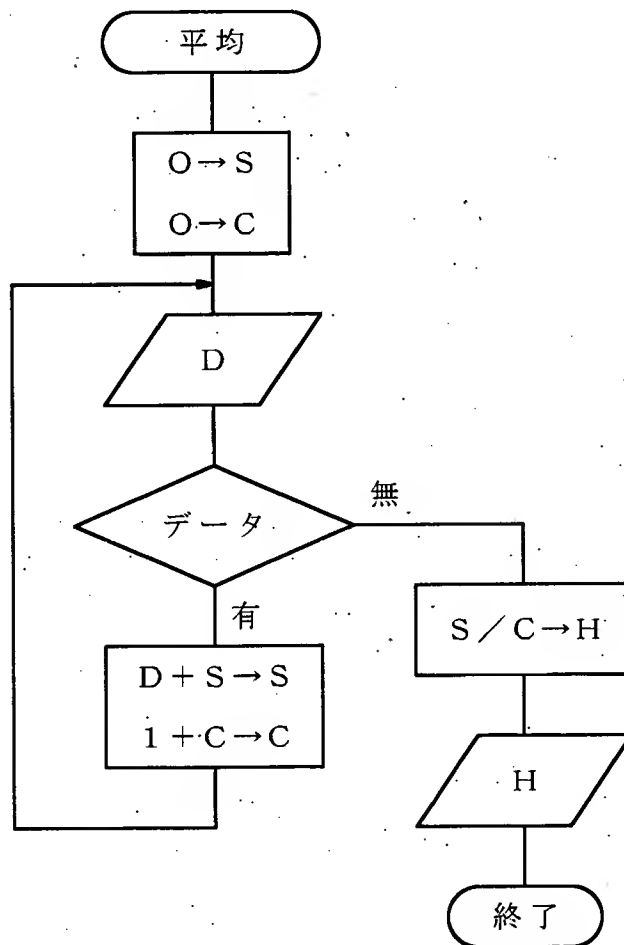
【図 29】

	時間1	時間2	時間3	時間4	時間5	時間6	時間7
CPU1	CPU1						
CPU2							
CPU3							
CPU4							

【図 3 0】

	時間1	時間2	時間3	時間4	時間5	時間6	時間7
CPU1							
CPU2	CPU1	CPU1	CPU2	CPU2	CPU2	CPU1	CPU1
CPU3							
CPU4							

【図 3 1】



【書類名】 要約書

【要約】

【課題】 縦書きプログラム流れ図等の不都合を解消し、且つ、プログラミングの手法を一変させるような、多次元の横書き流れ図を作成するための多次元プログラミング装置及び多次元プログラミング方法を提供することを目的とする。

【解決手段】 前記中央処理装置 9 A が、2 次元流れ図、3 次元流れ図、4 次元流れ図等の多次元の横書き流れ図を作成するためのオブジェクト情報が格納されたオブジェクト MD ファイル（オブジェクトファイル部 5）からオブジェクト MD（オブジェクト情報）を読み込む。次に、前記中央処理装置 9 A が、プログラマの入力指令に基づき、前記オブジェクト情報を用いて前記多次元流れ図を編集する。そして、その編集された前記多次元流れ図に基づき、多次元横書き流れ図を描画し、前記表示装置 9 2 に表示する。一方、前記中央処理装置 9 A は、編集された前記多次元流れ図を前記オブジェクトファイル部 5 に保存する。

【選択図】 図 8

特2001-315109

出 願 人 履 歴 情 報

識別番号

[500325908]

1. 変更年月日 2000年 7月11日

[変更理由] 新規登録

住 所 東京都大田区大森西4-10-11 田中マンション203

氏 名 嶺元 政輝